CMS IT Software Application Developer's Conference

September 2007

Agenda

- Introduction - Alan Constantian
- 3-Zone Architecture - Mark Hogle
- Java Development Standards - George Linares
- Security - Dick Lyman
- Enterprise Data Center - Sherry Wilke

3-Zone Architecture

September 2007

Agenda

- 3-Zone Architecture Overview
- 3- Zone Intrusion Detection and Intrusion Prevention
- Securing 3-Zone Access with IACS

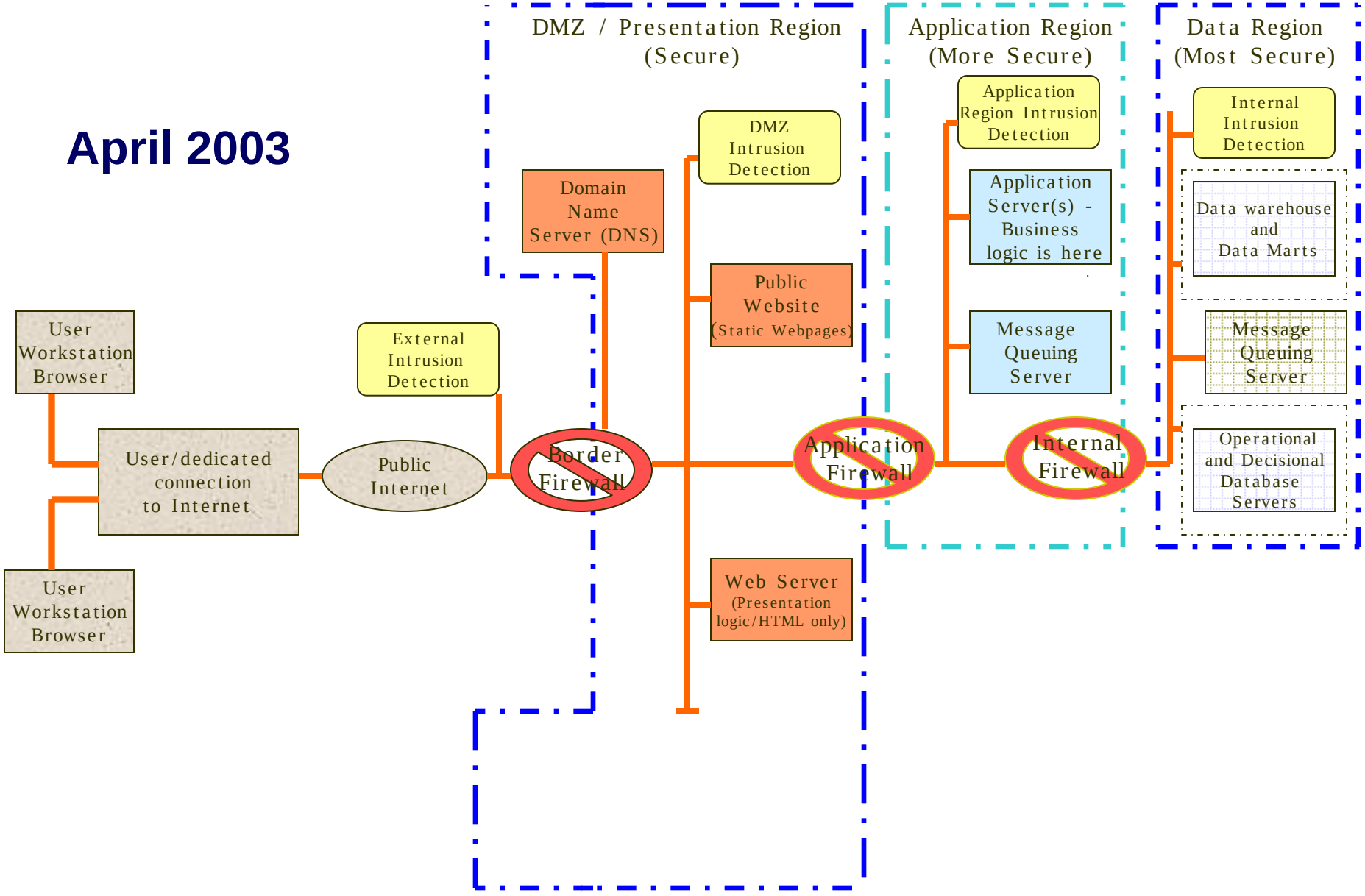
Background

- **3-Zone architecture established in 2002**
- **Associated standards established in 2002-2004 timeframe**
- **Separates software components into Presentation, Application and Data zones**

What is the CMS 3-Zone Architecture?

- **Describes the framework and infrastructure for mid-tier software applications**
- **Web-enabled infrastructure standard to be used for Internet, Extranet and Intranet applications.**
- **Separates software components into Presentation, Application and Data zones**
- **Zones are physically separated by firewalls for layered security**

April 2003


DMZ / Presentation Region (Secure)
Application Region (More Secure)
Data Region (Most Secure)
Application Region Intrusion Detection
Internal Intrusion Detection
DMZ Intrusion Detection
Domain Name Server (DNS)
Application Server(s) - Business logic is here
Data warehouse and Data Marts
Public Website (Static Webpages)
Message Queuing Server
Message Queuing Server
User Workstation Browser
External Intrusion Detection
Operational and Decisional Database Servers
User/dedicated connection to Internet
Public Internet
Border Firewall
Application Firewall
Internal Firewall
User Workstation Browser
Web Server (Presentation logic/HTML only)

Security Features

- **Layered security – multiple firewalls, different vendors**
- **Segmented security – physical separation (not just logical)**
- **Firewall configuration – deny all unless explicitly required**

Examples of 3-Zone Principles

- No port is opened through two consecutive firewalls
- Each component only communicates with components in an adjacent zone
- Firewalls have an implicit deny all
- No remote administration of firewalls via the Internet
- More details are defined in the standards documents

Existing CMS Standards

- CMS Internet Architecture
- CMS Target Architecture
- J2EE Application Development Guidelines
- CMS Enterprise File Transfer Infrastructure
- Many others exist and can be found at the following URL…

http://www.cms.hhs.gov/SystemLifecycleFramework/09_Standards.asp

CMS Standards Coming Soon

- **CMS Technical Reference Architecture (TRA) – master document**
- **Detailed supplements to complement TRA (e.g. supplement for security services, new java development standards, etc.)**
- **As these documents are finalized, they will be posted to the same URL listed on the previous slide…**

http://www.cms.hhs.gov/SystemLifecycleFramework/09_Standards.asp

Is It Only Three Zones?

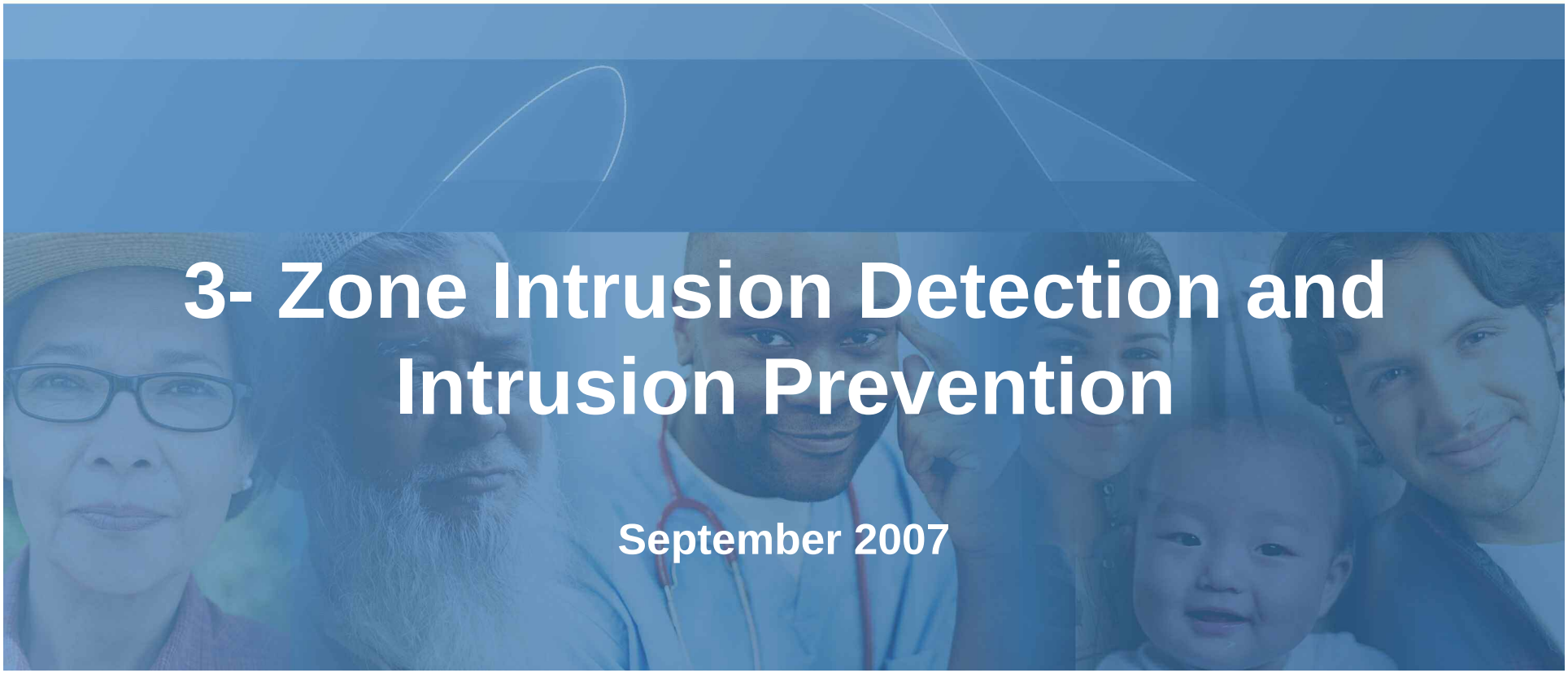
- CMS Definition of a Zone - network segment that is protected by a firewall or firewalls
- 3-Zone describes the application hosting environment experienced by the end user
- Multiple supporting zones or networks exist
- Application developers must understand the resources available to them
- Application developers do not access and interact with the production environment
- Application developers do access and interact with the development environment

Examples of Other Zones/Networks

- Administrative Management Network
- Firewall management network
- Security network (intrusion detection alerts and other related traffic)
- Backup network
- Access to these networks is protected by 2-factor authentication (something you have and something you know)
- Access to these networks is encrypted with virtual private network (VPN) technology

Other Considerations

- **CMS Internet Architecture is a very high-level document**
- **Actual implementation involves much more detail**
  - SSL encryption is not performed by web servers
  - SSL accelerators and load balancers front-end the 3-Zone
  - XML firewalls are used to secure web services
- **Developers need to understand how to access and deploy code to the 3-Zone**
- **Clear understanding of CMS configuration management processes is required**
- **Developers must build production scheduling needs into project plans**



3- Zone Intrusion Detection and Intrusion Prevention

September 2007

What is Intrusion Detection?

- **Intrusion detection is the process of monitoring the events occurring in a computer system or network and analyzing them for signs of intrusions.**
- **Intrusions are defined as attempts to compromise the confidentiality, integrity, availability or to bypass the security mechanisms of a computer or network.**
- **Intrusion Detection Systems (IDS) can be software or hardware products that automate the monitoring and analysis process.**
- **Intrusion Prevention Systems (IPS) respond proactively to intrusions to automatically thwart attacks in real-time**

Why We Use IDS/IPS?

- Monitor and analyze user and system activities
- Analyze system configuration and vulnerabilities
- Assess system and file integrity
- Recognize patterns of typical attacks
- Analyze abnormal activity patterns
- Track user policy violations

Types of IDS: Network-Based IDS

- Two principal types of IDS are host-based and network-based.
  - Network-Based IDS (NIDS)
  - NIDS sensors are placed at various points in a network.
  - NIDS are implemented as protocol analyzers with the capability to recognize particular events, scrutinizing the traffic for signs of attack or penetration attempts.
  - NIDS generally rely on predefined “attack signatures” to detect and identify attacks. When the IDS detects a series of events that matches the attack signature base it notifies the network administrator

Types of IDS: Host-Based IDS

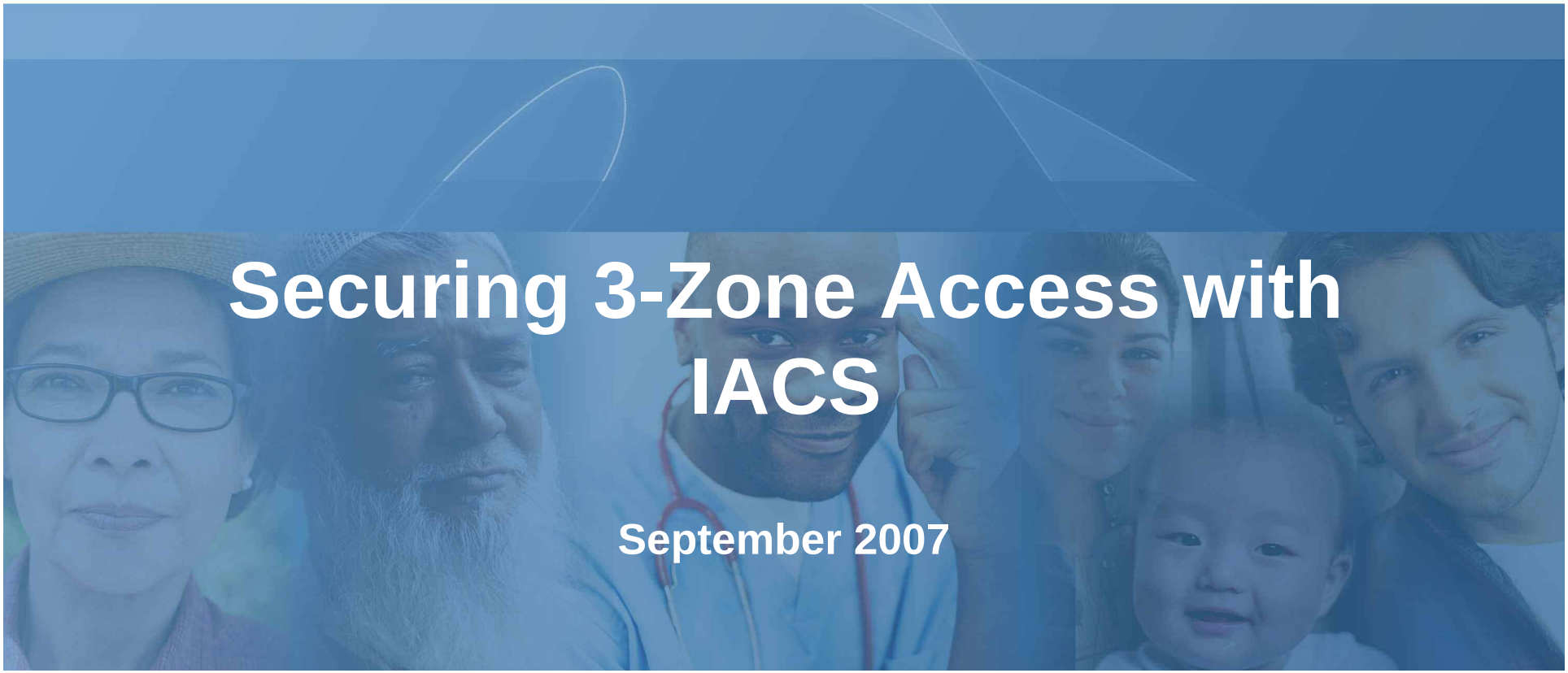
- **Host-Based IDS (HIDS)**
  - **Operate on information collected from within an individual computer system. Must be installed on each individual computer system that is to be monitored or protected.**
  - **HIDS are useful when most of the network traffic to and from the web components are encrypted.**
  - **HIDS normally utilize information sources of two types, OS audit trails and system logs. OS audit trails are usually generated at the innermost kernel level of the OS and are therefore more detailed and better protected than system logs.**
  - **Can monitor the overall interaction between use and application, tracing unauthorized activity to individual users.**

Deployment Strategy

- **Generally Sensors will be placed at the following locations:**
  - **Perimeter (Internet)**
  - **Sensor in every zone (3-Zone Architecture)**
  - **Third Party Connections**
  - **Critical Resources (e.g. sensitive data stores)**
  - **Remote Access Entry Points**

Key Concepts

- **Know Your Environment: A thorough understanding of the organizations enterprise must be attained prior to implementing an IDS solution**
- **Baseline normal network traffic and service activity and note anomalies**
- **Tune sensors - 3-Zone architecture makes this easier due to well-defined traffic**
- **Monitor 24x7 - enterprise approach of pooling resources becomes critical**



Securing 3-Zone Access with IACS

September 2007

IACS Topics

- What is IACS?
- IACS Key Concepts
- Current Development Underway
- Why Do You Care?

IACS – What is it?

- **IACS is the CMS centralized identity and access management system**
- **IACS is the CMS implementation of SUN Identity Manager (IM), Access Manager (AM) and Directory Server products and provides the following:**
  - A Single User Identity for CMS application users
  - Reduced cost of enterprise security services across multiple applications
  - Access to CMS systems via a user-friendly web interface
  - Paperless process that streamlines the access to CMS applications using:
    - Electronic terms & conditions
    - Electronic privacy statement

IACS Components and Functions

- **IACS Identity Manager (IM) provides:**
  - User Registration
  - Approval/Provisioning Workflows
  - Support of various system resources
- **IACS Access Manager (AM) provides:**
  - Authentication Service
  - Not currently used by Fee for Service users
- **IACS Directory Server (aka LDAP)**
  - Directory store of user information (name, address, userid, password, etc.)
  - Kind of like the telephone book

Functions of Identity Management

- Automated user provisioning to improve operational efficiency and enhance security
- Secure, automated password management to improve service levels and lower costs
- User self-service and delegated administration to lower support costs
- Automated data synchronization to lower workloads associated with handling change
- Comprehensive auditing and reporting to improve security compliance

IACS – Integrated Self-Service

- **Self-Registration process**
  - With full workflow/approval enablement
- **Single-point end-user account self-service**
  - Basic account self-service and attribute management
  - Single-point password sync/reset
  - Integrated challenge/response for forgotten passwords
- **Integrated workflow, approvals & audit**

CMS User Account Lifecycle – IACS

- A new user accesses the IACS system to requests a user account, the user fills out a web form.
- The user request flows through a work flow approval process.
- Upon approval the IACS system provisions the user account to various resources within CMS including (Enterprise Directory, RACF, DB2, and Active Directory).
- User must certify their account annually
- User account is archived when no longer required

IACS Role Types

- **End User**
  - End User of Fee for Service Applications
- **EPOC (External Point of Contact), Security Official (SO) or UGA (User Group Administrator)**
  - CMS delegated Administrator
  - Responsible for ensuring the end user's identity
- **CMS Authorizer**
  - CMS Business Owner
  - Responsible for ensuring the identity of the EPOC
- **Help Desk User**
  - Customer Support Representative
  - Responsible for assisting end users with password administration

IACS Key Concepts

- IACS is **wide**, not **deep**
- Well-defined **user communities** must be established
- There will be an agreed upon, well-known **Chain of Trust** for each Community

Wide, Not Deep

- **IACS core functions cover the gamut**
  - User registration, core workflows, password resets, etc.
- **Application-specific data is stored outside of IACS**
  - A Medicare provider database may be useful, but would not be housed by IACS
- **Depth of IACS is Community dependent**
  - There is no universal answer
- **Automated Approver**
  - A mechanism for automatically approving end user registration outside of IACS
  - Approval requests are sent to this system in lieu of a person
  - IACS calls or accesses the external system and trusts it
  - DME CBSS is an example

Establishing Communities

- Participate with OIS and Business community to define Communities (e.g. FI, Carrier, Provider)
- Use business process models to define business risks and required trust levels per community
- Establish assurance levels required for that trust
- Define rules for that community
- A business owner must be established for each community

Chain of Trust

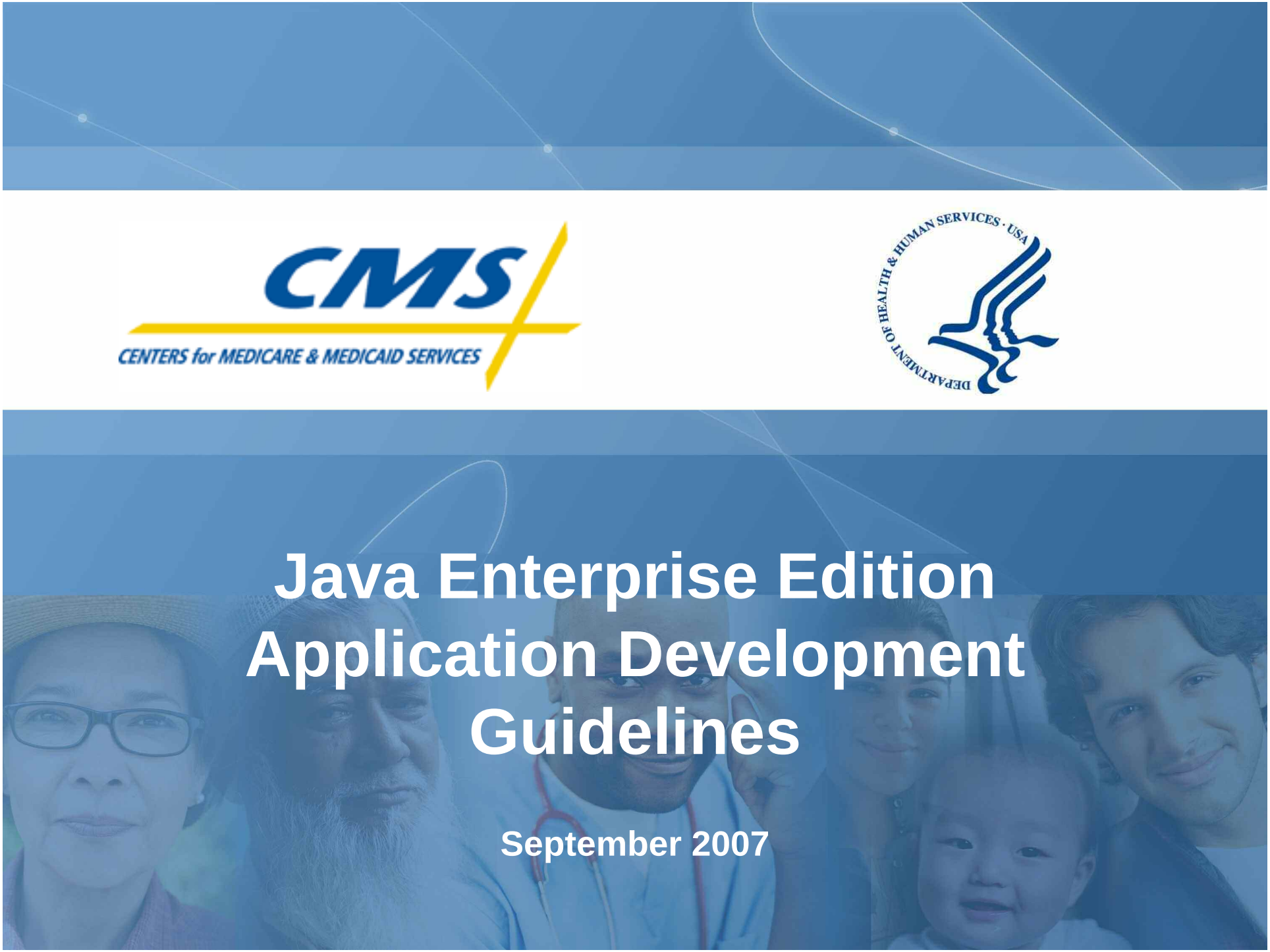
- Define and Publish the process to vet people into the community
- Others need to understand the process so they can trust the community
- Communities become building blocks that business owners can use to allow people into the applications

Current Development Underway

- IACS has been in production since the fall of 2005
- Many enhancements and added functionality over the past two years
- Medicare Advantage and Part D user community was the first operational community
- Currently building functionality to service the provider community and the FI/Carrier/MAC community

Why Do CMS Application Developers Care?

- No longer need to develop stove-piped, application-specific identity and access management solutions
- Need to understand how IACS works and how to integrate with it
- Some applications will require additional work if application-specific needs are not covered by IACS
- Need to engage IACS team early to communicate requirements
- Understanding end users and their sponsors is critical



Java Enterprise Edition Application Development Guidelines

September 2007

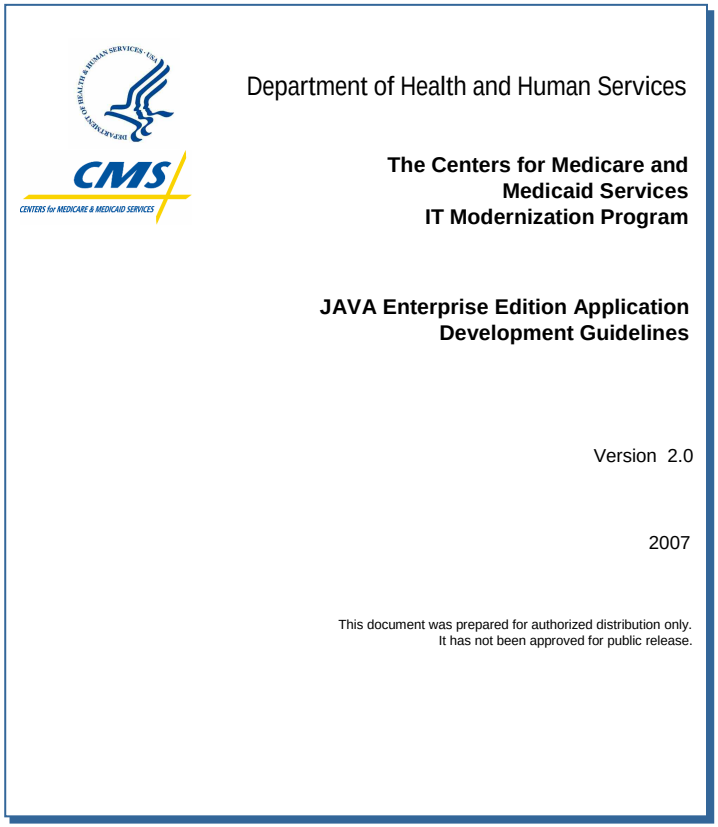
Agenda

- **Purpose of the *Java EE Guidelines***
- **Complying with the CMS Three-Zone Internet Architecture**
- **Application Security**
- **Messaging Services and Concepts**
- **Infrastructure Services**
  - Components
  - Testing, Validation and Production Environments
  - Support
- **Questions**

Purpose of the CMS *Java EE Guidelines*

- **Provide clear guidance to Java EE software developers on CMS standards**
- **Summarize key lessons learned and research into industry best practices on Java EE-based enterprise application development as they apply to CMS**
- **Address compliance with CMS three-zone architecture**
- **Current Guidelines are posted at**

http://www.cms.hhs.gov/SystemLifecycleFramework/09_Standards.asp



Department of Health and Human Services

The Centers for Medicare and Medicaid Services
IT Modernization Program

JAVA Enterprise Edition Application Development Guidelines

Version 2.0

2007

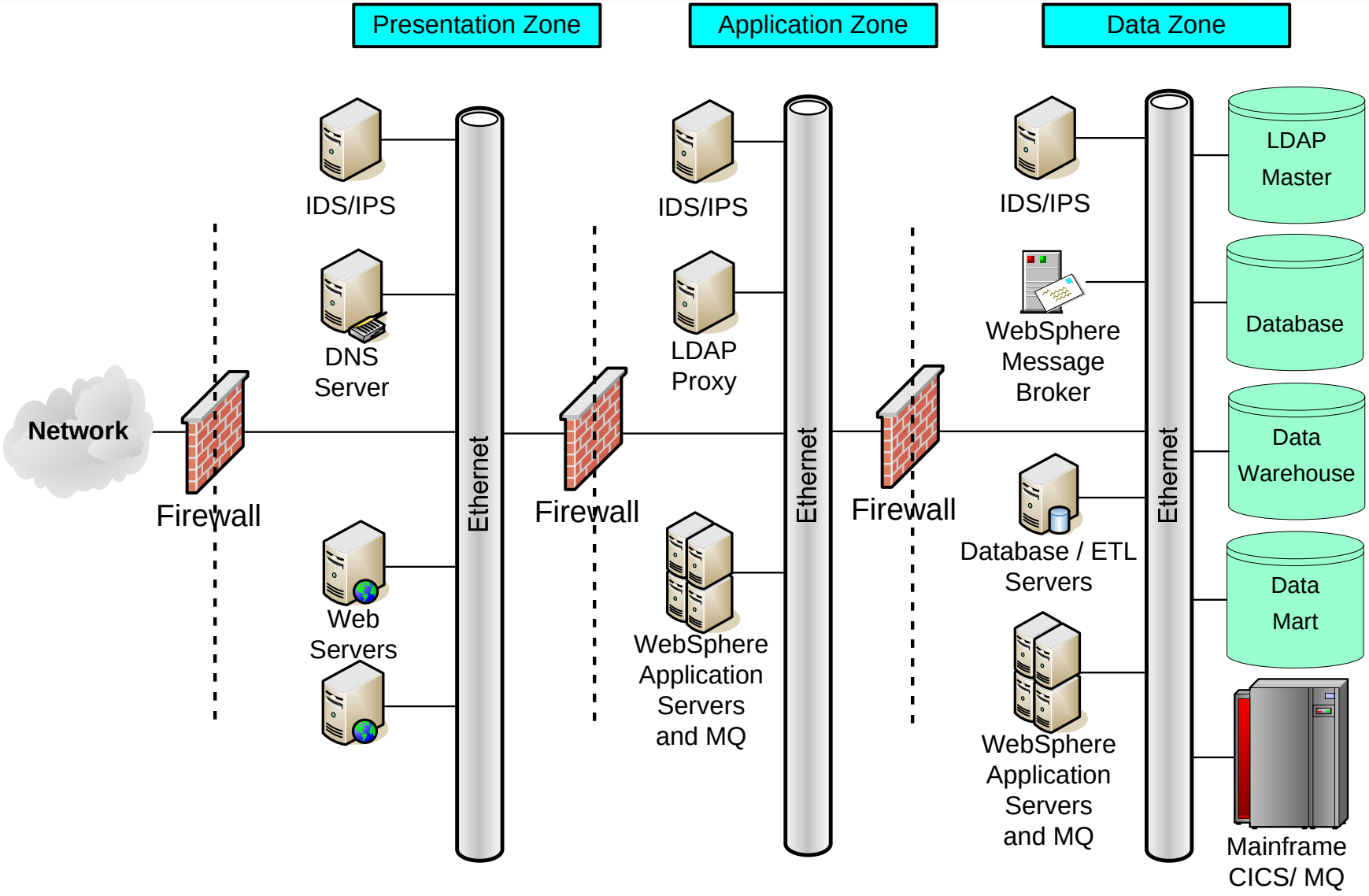
This document was prepared for authorized distribution only.
It has not been approved for public release.

The Java EE Guidelines document (formerly the J2EE Application Developments Guidelines) is one in a series of technical guidelines currently being updated for the CMS developer community

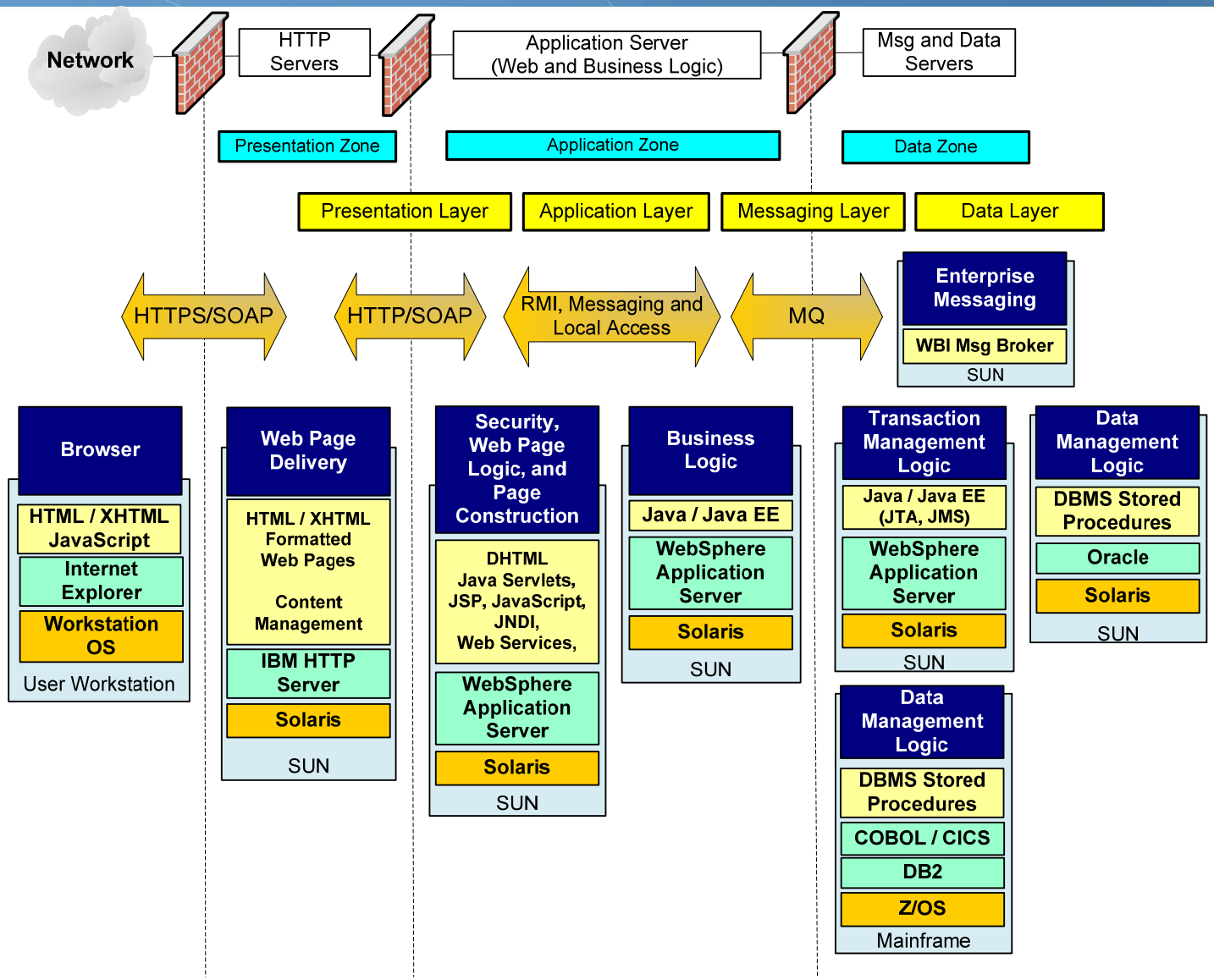
CMS Java EE Architecture Considerations

- **Promote the use of industry best practices**
  - Good object-oriented design promoted by the use of proven design patterns
  - Service-Oriented Architecture Principles
  - An open architecture, independent of underlying object models
- **Use layered architecture for flexible development & deployment of solutions**
- **Ensure scalable architectures**
- **Develop Failure-Resilient & High-Availability Solutions**
- **Provide support for transaction management**
- **Apply end-to-end management and security**
- ***And*** **build applications that are easy to test and maintain**

Physical View of CMS Three-Zone Internet Architecture



Application View of the CMS Three-Zone Architecture



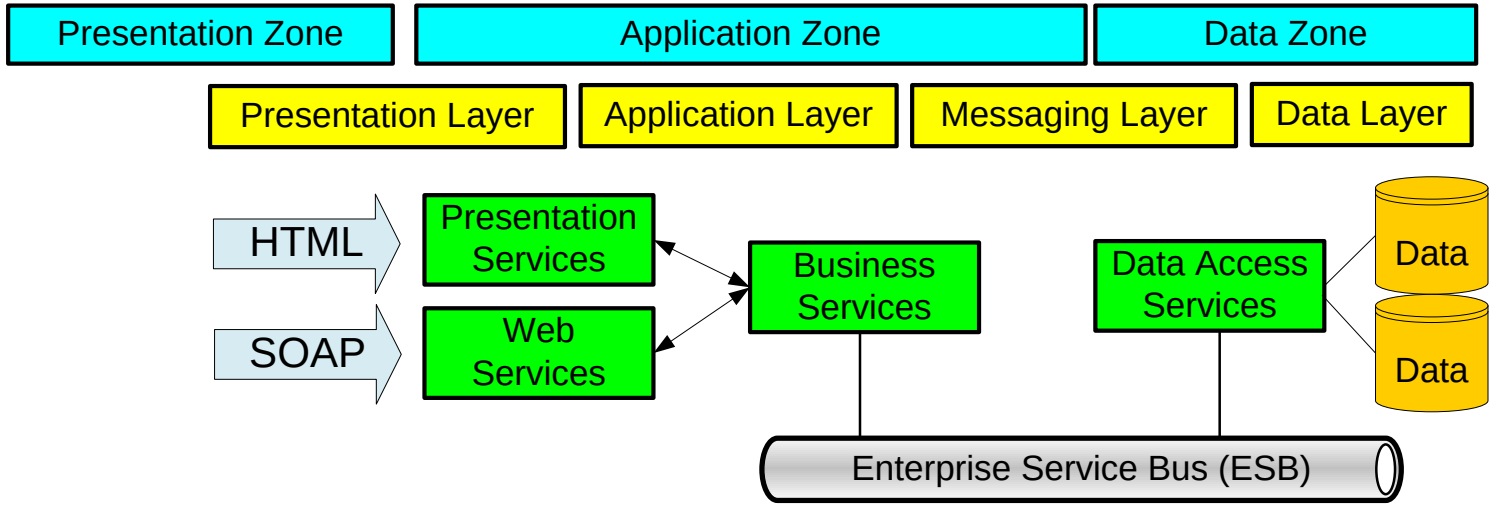
Java EE Application Architecture

Java EE applications consist of components in loosely coupled layers

- **Presentation Layer**
  - Dynamic and Static Web pages (Servlet, JavaServer Pages (JSP), or JavaScript)
  - Application Authentication
- **Application Layer**
  - Business components that run on the Application Server
- **Messaging Layer**
  - Physical Data Transport between the Application Layer and Data Access Service Layer
  - Implemented with WebSphere MQ and WebSphere Business Integration Message Broker
- **Data Access Layer**
  - Components that abstract and encapsulate access to the Data Zone
  - Encompasses Business Rules surrounding the data
  - Role-Based security (Only necessary data should be sent)
- **Java EE components are compiled and assembled into a Java EE application, and then are run and managed by the Java EE server**

Service-Oriented Architecture Principles

- **Services can be Web Services… OR NOT!**
- **Services should be designed within a business process context.**
- **Services should comply with standards (from industry, government, and CMS) wherever possible.**
- **Services should encompass a complete and independent unit of work.**
- **Services should be modular, reusable, complete, error-tolerant, and sufficiently coarse-grained to be requested individually or as a part of a larger composite service request.**
- **Services should be "loosely-coupled"**
- **Services must expose a stable interface, or service contract, that is independent of the internal code implementation, provides a layer of abstraction from the internal business logic, and ensures an appropriate level of security validation.**



Data Access Services

- **CMS Data Access Services include:**
  - Application access to CMS data stores
  - Data abstraction and encapsulated details of database structures
  - Data management and data optimization activities
  - Independent construction and maintenance of services without impact to the applications that use them
- **Supporting Services include**
  - Security and privacy control
  - Error handling
  - Auditing

Application Security

Application Development Perspective

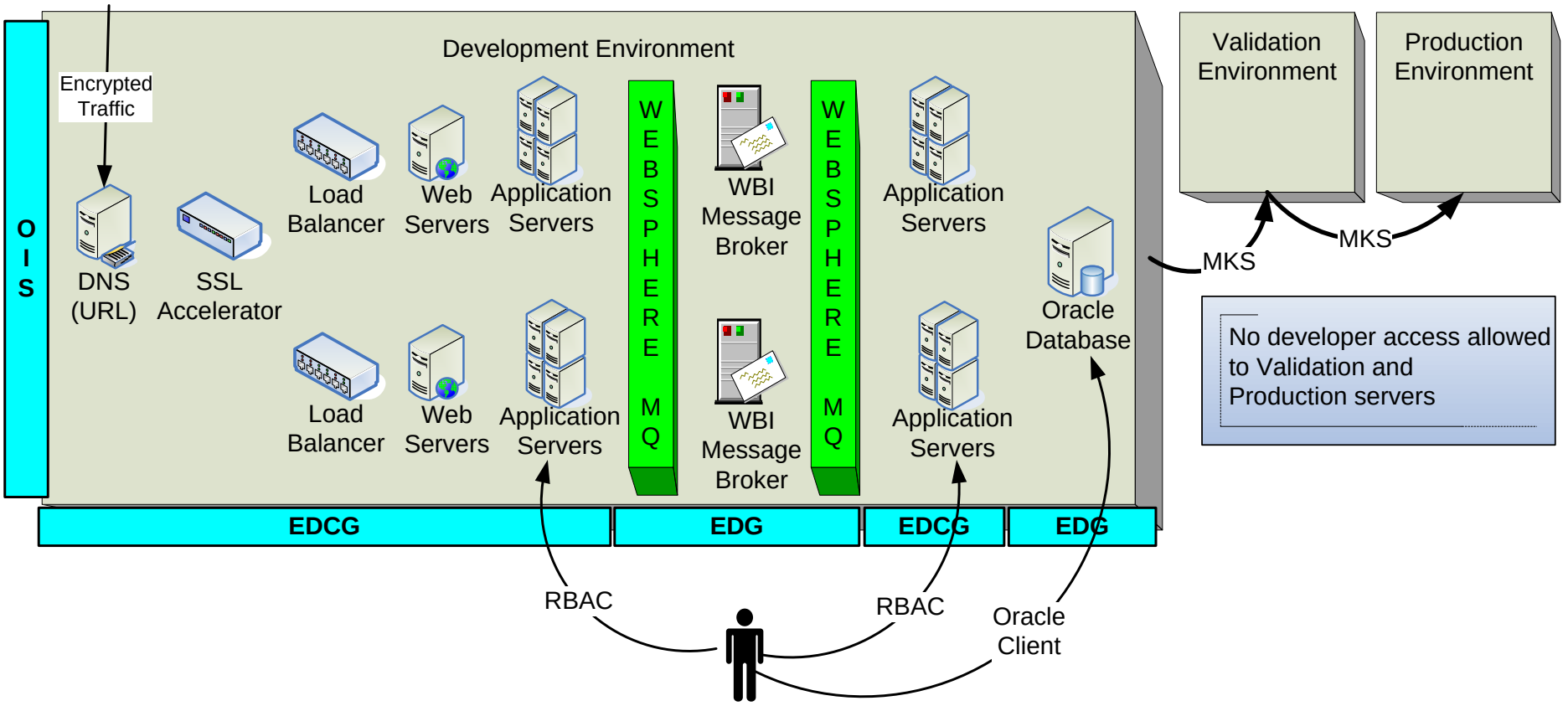
- **Applications must integrate with “IACS”, which provides**
  - User Registration and Provisioning
  - Application Authentication / login page
- **Applications must provide Authorization / Role-Based Security**
- **Applications may utilize Java specifications for Java Naming and Directory Interface (JNDI) and Java Authentication and Authorization Service (JAAS) to further integrate with IACS via secure LDAP**
- **Source Code Security Analysis & Review (using Fortify SCA)**
- **Website Vulnerability Analysis and Quality Review (using Watchfire WebXM and AppScan)**

Messaging Layer

- **Messaging will be used for**
  - Communicating among individual applications and/or services
  - Synchronous inter-process communication, e.g., calling a service and waiting for a response
  - Asynchronous communication, e.g., sending an email
- **WebSphere MQ will be used to communicate and invoke Data Zone Software Components / Services**
  - Standard message body should be constructed as XML document
  - Data Zone clustering
- **WebSphere Business Integration (WBI) Message Broker is used as CMS's Enterprise Service Bus, which provides:**
  - Ensures that Valid Messages and/or Service Requests are made
  - Message Transformation
  - Message Routing
  - Application E-mail services

Infrastructure Services

Typical Infrastructure Setup



Infrastructure Services

Development and Environments at CMS

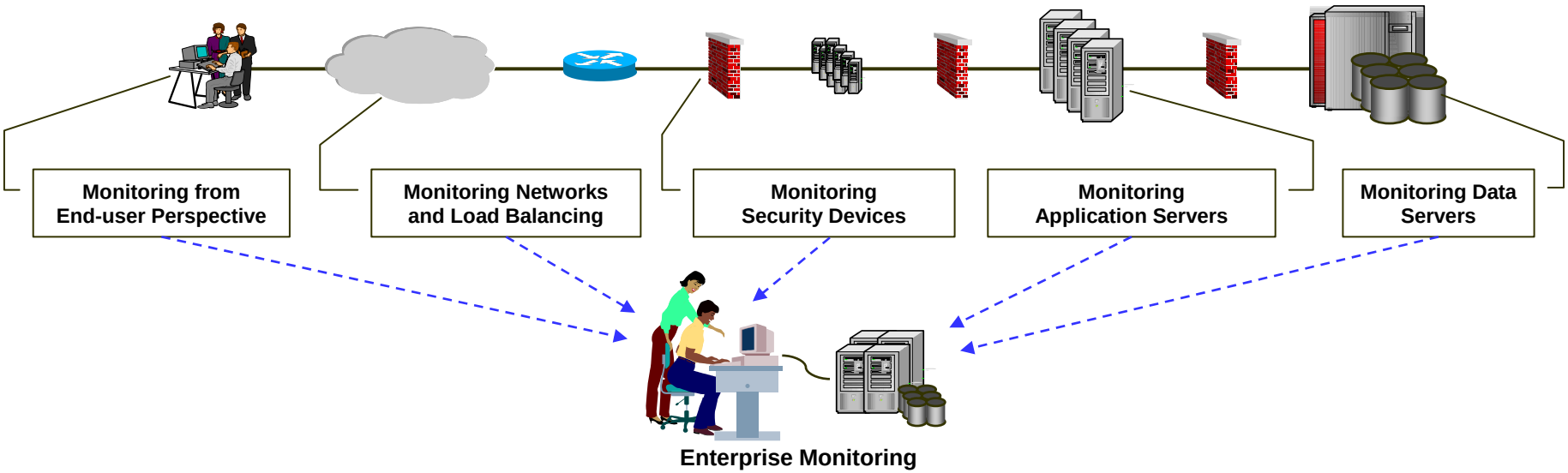
- **Development and Design**
  - Local contractor-based Development Environments (off-site)
  - Eclipse-based IDEs
  - Unit Testing considerations
  - Design should take into account
    - 3-Zone architecture
    - Security
    - High-availability – both vertically and horizontally (impacts object interfaces and message to queue mapping)
    - Capacity and performance

- **Environments at CMS**
  - Development* (for system and integration testing)
  - Validation (for user acceptance testing and performance testing)
  - Production
  - All environments look the same and have the same components
  - Production environments configured for high-availability (multiple JVMs, clustering, etc)

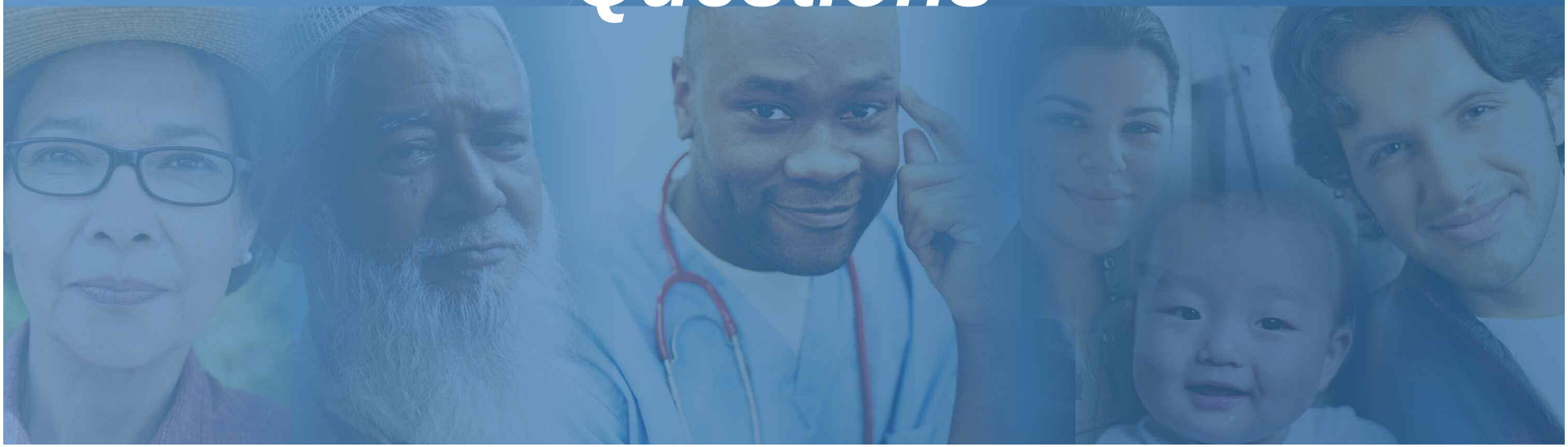
Production Considerations

- **Production Environment**
  - Application Server and Messaging Monitoring
  - Network & System Management and Monitoring
  - Incidents & Problem Tracking
  - Performance Management & Capacity Planning
- **Life-Cycle Support**
  - Configuration & Change Control
  - Deployment / Release Management (MKS)
  - MQ / WAS Administration
  - Operations Management (CITIC)
  - Enterprise Database Support

Production Infrastructure Resources



Questions



CMS Application Security

September 2007

Agenda

- Information Security
- An Effective Security Strategy
- CMS Security Flows
- Application Security
- Why We Do Security
- Questions?

Information Security

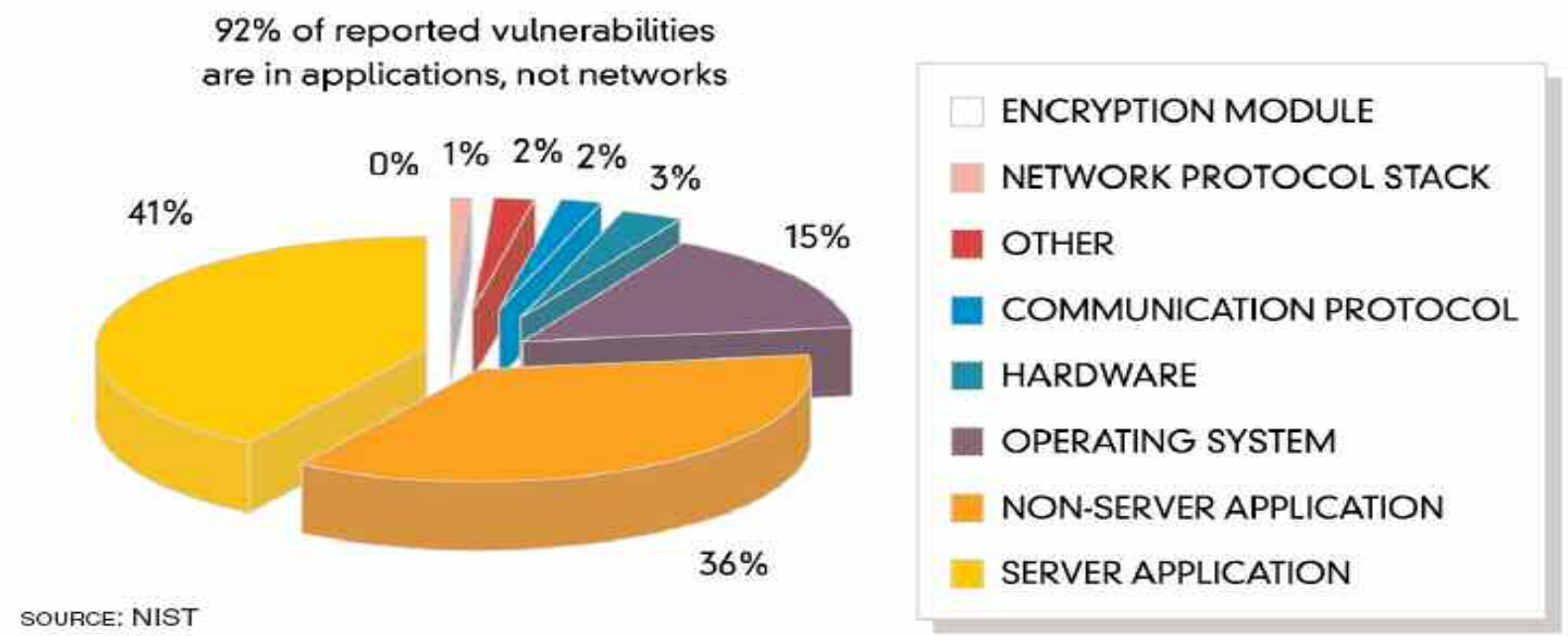
Information security cannot stand on its own: it relies on good network, application and host security. Therefore, the security strategy outlined within the proposals must reflect technologies and controls within this logical view of security.

An Effective Security Strategy

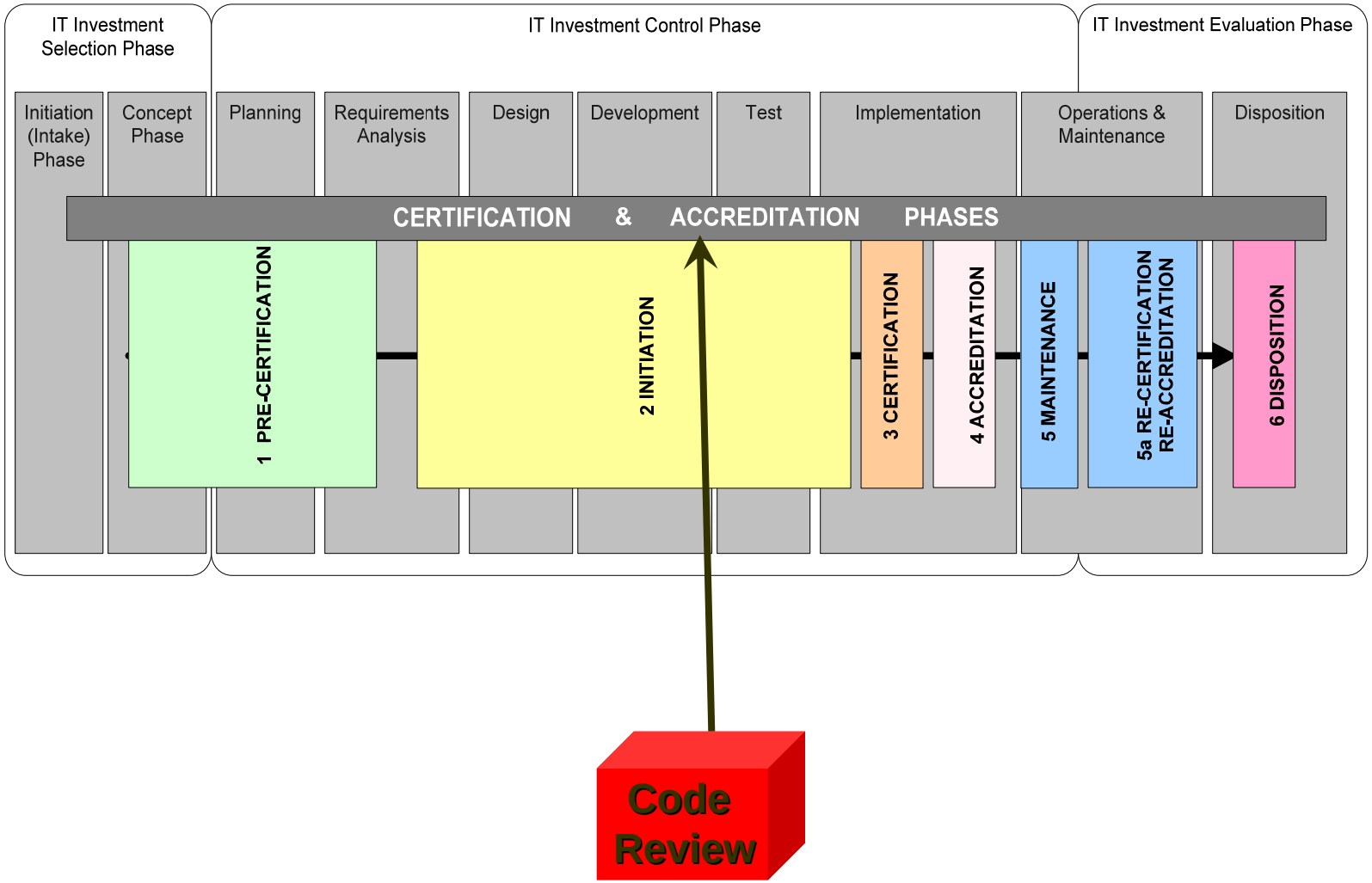
- Policy enforcement
- Peripheral security
- Central security management
- Identity & access management
- Physical and environmental security

A Few Facts and Figures:

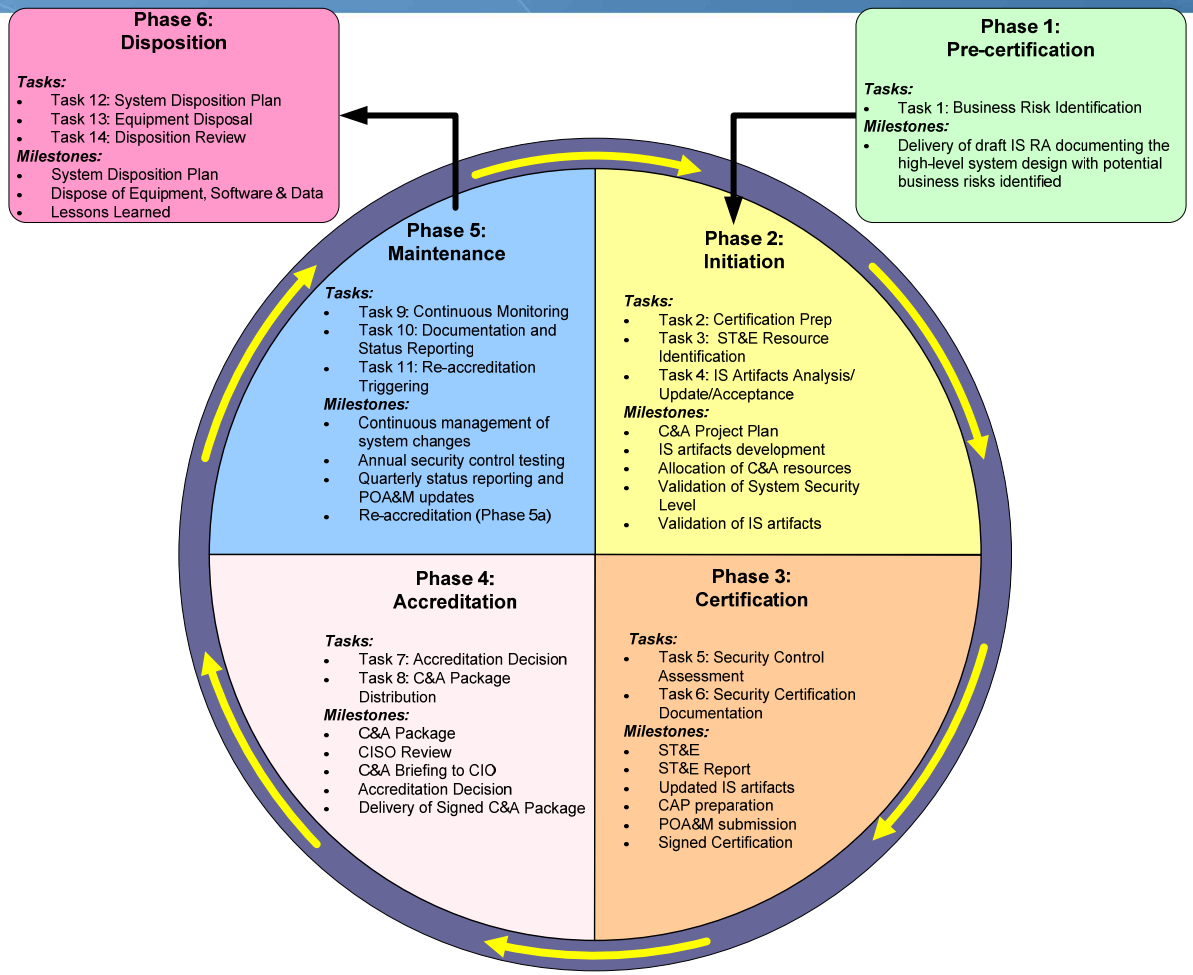
How Many Vulnerabilities Are Application Security Related?



CMS Framework and C&A Phases



CMS' C&A REQUIREMENTS

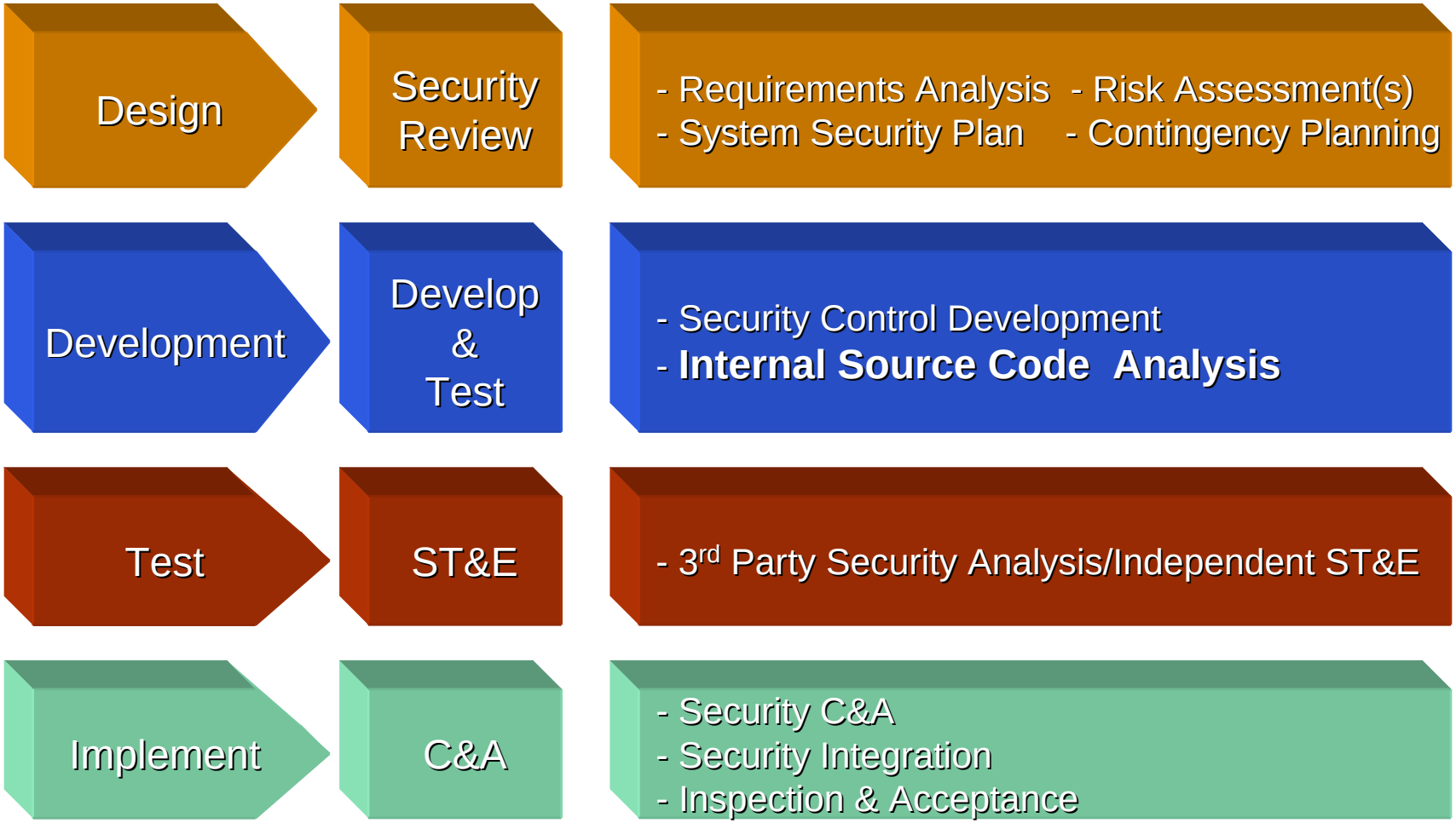


Security Program Documents

- **Updated to meet**
  - OMB Directives
  - NIST Requirements
  - HHS Directives
  - CMS CIO Directives
- **Streamlined the CMS C&A process**
- **Aligned with the HHS Enterprise Program Life Cycle**
- **Primary Driver – NIST SP 800-53 Rev. 1 Controls and NIST SP 800-53A testing requirements**

http://www.cms.hhs.gov/informationsecurity

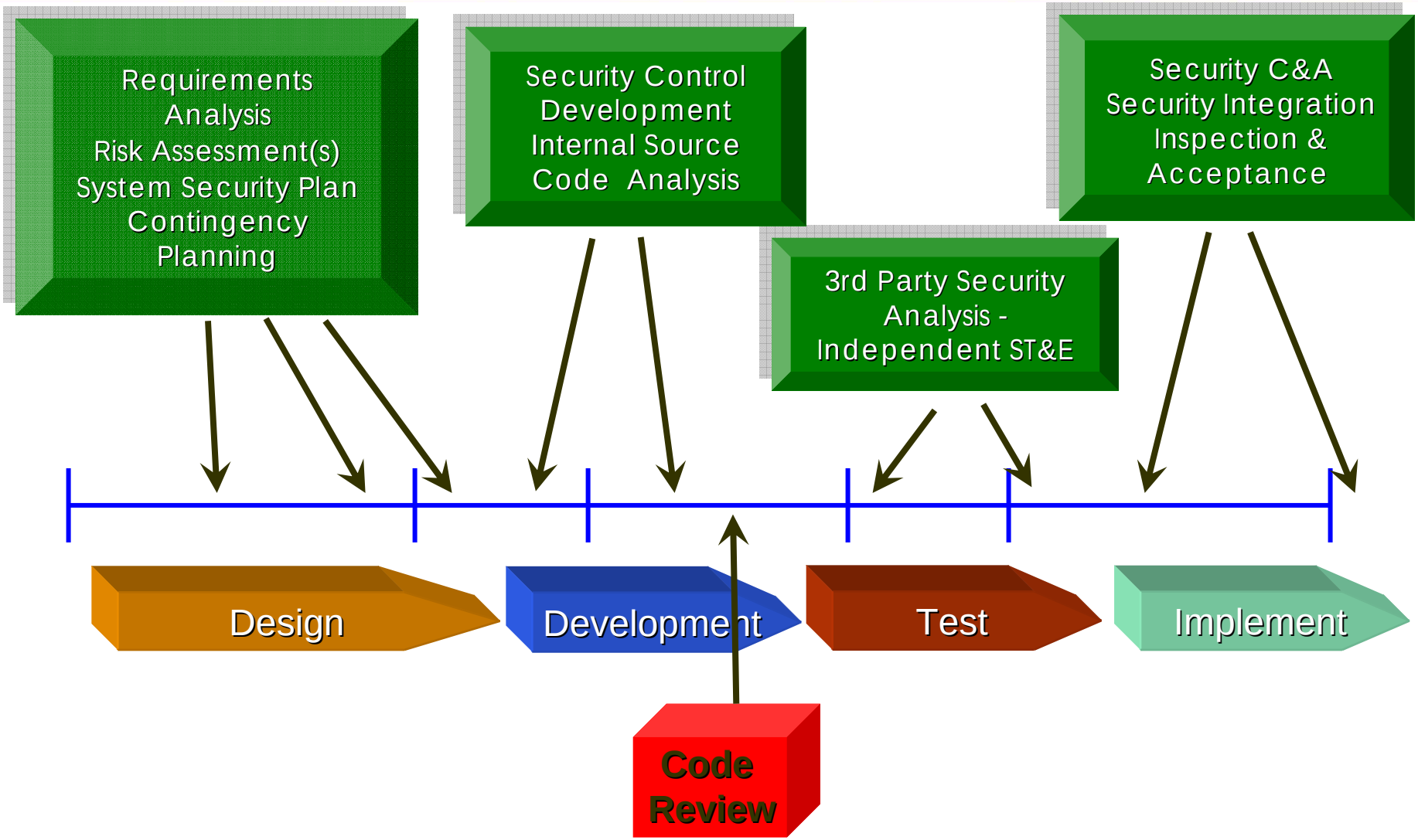
CMS Security Flows



Why Do Application Security?

- Security vulnerabilities in applications are to be considered bugs, the same way as functional bugs, and tracked in the same manner.
- Consider security as added value in an application.
- $1 spent up front saves $10 during development and $100 after release*

Software Security Tollgates



Software Vulnerability Analysis

- Find flaws in the source code early
- Many different techniques
  - Static (against source or compiled code)
    - Security focused static analysis tools
    - Peer review process
    - Formal security code review
  - Dynamic (against running code)
    - Scanning
    - Penetration testing
- Goal
  - Ensure completeness across all vulnerability areas
  - Ensure accuracy (minimize false alarms)

Application Security Testing

- Identify security vulnerabilities during testing
  - 3rd party
  - Independent
- Develop security test cases
  - Based on requirements
  - Be sure to include “negative” tests
  - Test all security mechanisms and common vulnerabilities
- Flaws feed into defect tracking and root cause analysis

Application Security Defect Tracking and Metrics

- ***"Every security flaw is a process problem"***
- ***Tracking security defects***
  - *Find the source of the problem*
    - *Bad or missed requirement, design flaw, poor implementation, etc…*
- ***CAP/POAM***
- ***Metrics***
  - *What lifecycle stage are most flaws originating in?*
  - *What security mechanisms are we having trouble implementing?*
  - *What security vulnerabilities are we having trouble avoiding?*

Configuration Management and Deployment

- Determining / documenting the baseline configuration
- Ensure the application configuration is secure
- How do you control and audit this data?
  - Design configuration data for audit
  - Put all configuration data in Configuration Management
  - Audit configuration data regularly
  - Don't allow configuration changes in the field

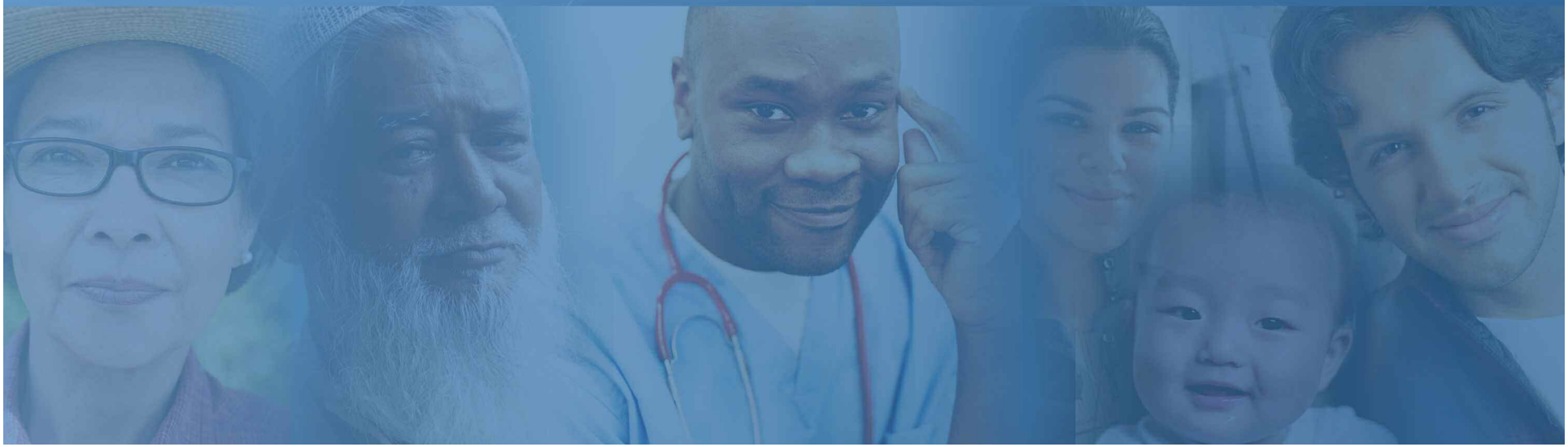
In Summary, We Do Security To…

Enable Business Integrity

&

Maintain Assurance Levels

Questions?



CMS Enterprise Data Centers

September 2007

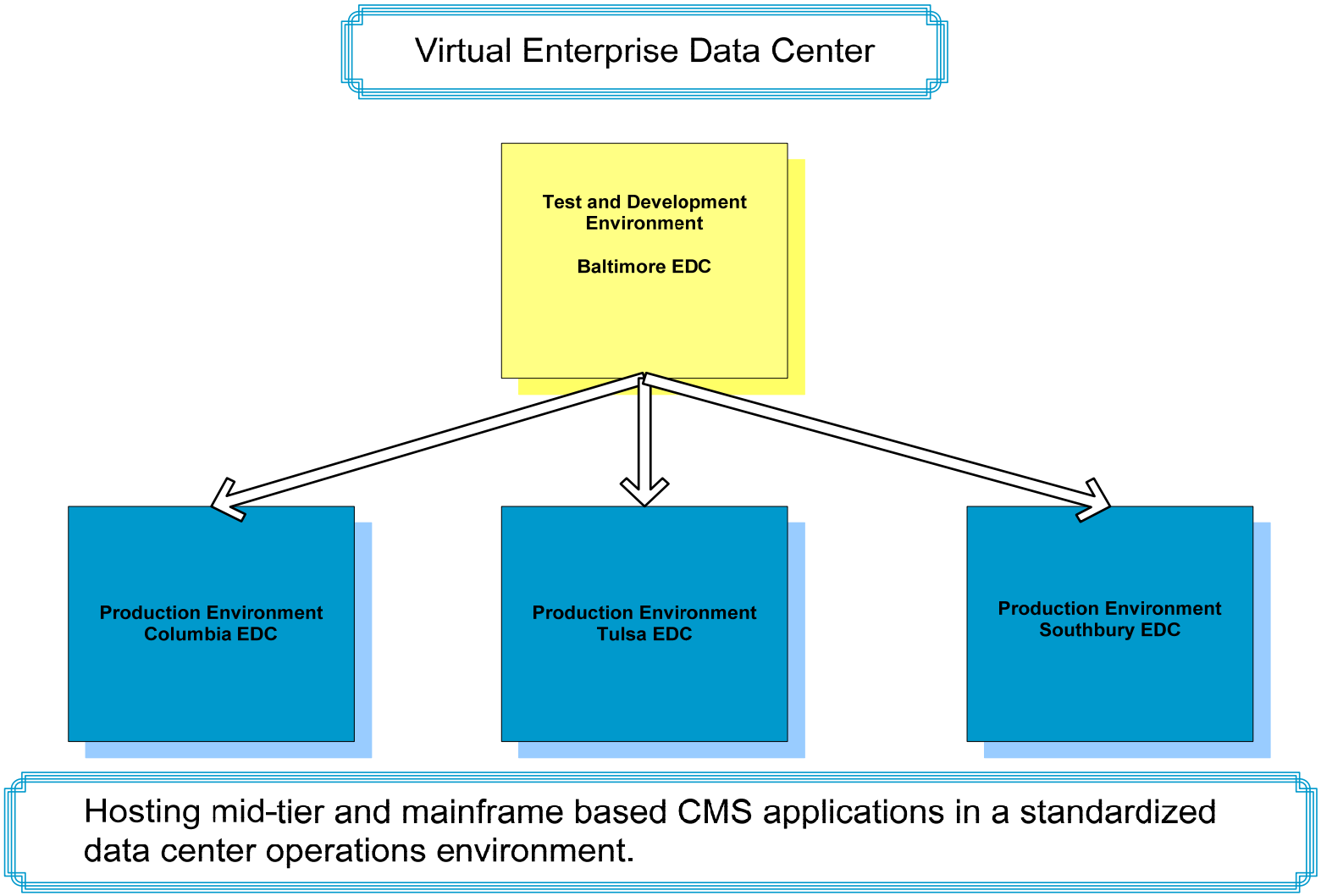
Agenda

- Enterprise Data Center (EDC) Goals
- Virtual EDC
- EDC Standard Services
- EDC Validation Environment

Enterprise Data Centers Goals

- Host secure, reliable mission-critical applications
- Provide greater control over security and privacy
- Scale EDC operations as volumes grow
- Manage traditional mainframe data center operations and new mid-tier web application-hosting environments
- Enhance data center infrastructure as business requirements change and technology improves
- Improve service levels to beneficiaries and providers through support of web-based services

CMS Virtual Enterprise Data Center



EDC Standardization Benefits

- Facilitate the promotion of application releases from T&V environment through the standardization of hardware and software
- Improve the ability to implement new applications or make changes to existing applications across any EDC through the standardization of enterprise-level infrastructure
- Where an application is hosted will be transparent to end users
- Access any CMS production application in the virtual EDC using the IACS provisioned ID

EDC Standards - Software

- **Mainframe**
  - z/OS
  - RACF - Security
  - Tivoli Workload Scheduler – Enterprise Job Scheduler
  - WebSphere MQ
  - CICS
  - DB2
  - Enterprise COBOL
  - Endevor – Software Configuration Management

EDC Standards - Software

- **Mid-Tier**
  - Solaris
  - Tivoli Workload Scheduler – Enterprise Job Scheduler
  - WebSphere Application Server
  - WebSphere MQ
  - WebSphere Business Integration Message Broker
  - Oracle
  - MKS Integrity – Configuration Management Software
  - JAVA 2 Enterprise Edition (J2EE)

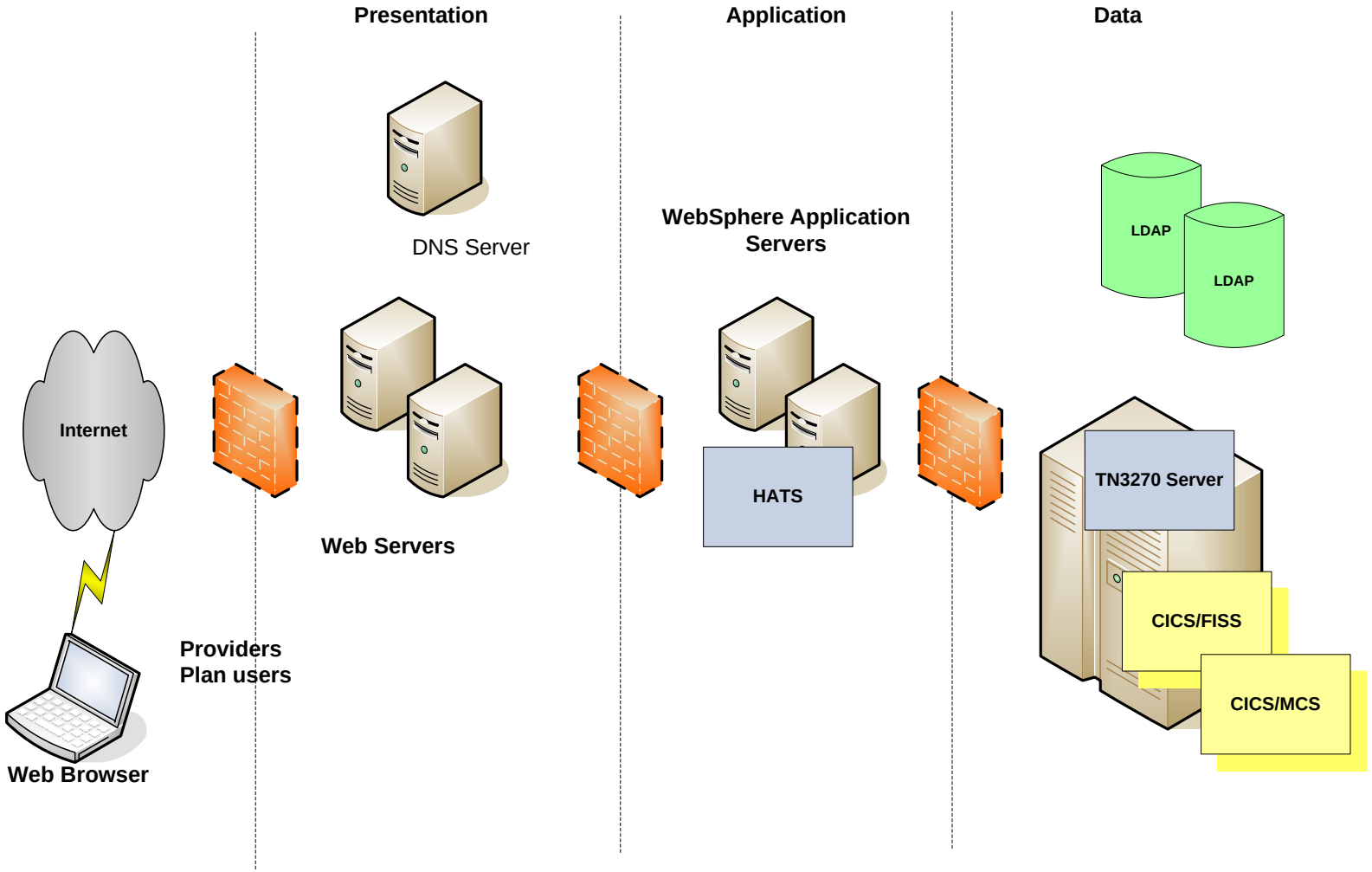
EDC Standard Services

- **Domain Name Services**
- **File Transfer**
  - Connect:Direct
  - Gentran
- **Host Access**
  - TN3270
  - HATS?
- **IACS**
  - Identity Manager
  - Access Manager
  - Directory Server (LDAP)
- **Messaging**
  - WebSphere MQ
- **Mail**

Enterprise Management Standards

- **Change Management**
  - Remedy
- **Problem Management**
  - Remedy

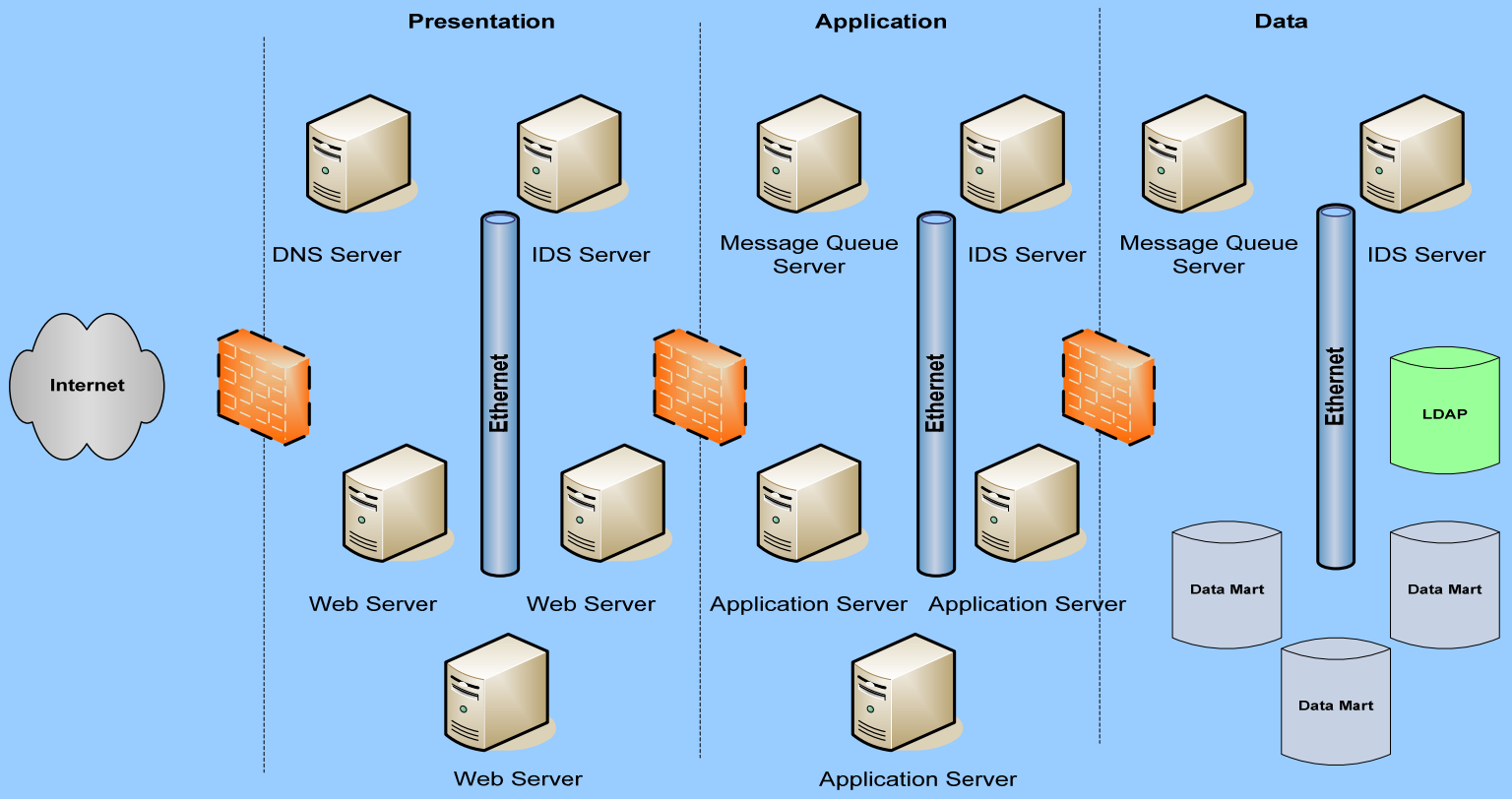
Host Access using HATS



EDC Validation Environment

- The validation environment will be built to production EDC standards in order to provide for the reliable validation of production releases
- Security standards will apply to the EDC validation environment just as they apply in production
  - Application support access will be highly restricted
- Adherence to both Change and Configuration Management practices is required

EDC Three-Zone Architecture for CMS Applications



- Ensure applications adhere to CMS Target Architecture
- Ensure applications adhere to CMS Internet Architecture

Some key indicators of an “EDC ready” application

- Adheres to CMS Target and Internet Architecture
- Can be scheduled using an automated tool (TWS)
- Has documented interfaces
- Requires minimal exception processing
- Code is under software configuration management
- Has a user community workflow available in IACS
- Adheres to the CMS Systems Development Life Cycle

CMS

CENTERS for MEDICARE & MEDICAID SERVICES

DEPARTMENT OF HEALTH & HUMAN SERVICES · USA